

FFmpeg

Reto Kromer • AV Preservation by reto.ch

Initiation à FFmpeg
Lichtspiel / Cinémathèque Berne
13 février 2026

1

Interface utilisateur

- cartes perforées et imprimantes matricielles
- interface graphique
anglais: graphical user interface (GUI)
- ligne de commande
anglais: command-line interface (CLI)
- interfaces sans contact

2

ASCII (1977/1986)																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
	Changed or added in 1963 version															
	Changed in both 1963 version and 1965 draft															

source: wikipedia.org

3

Structure de commandes Unix/Linux

\$0 **\$1** **\${n}**
command argument_1 ... argument_n

Syntaxes habituelles des arguments:

--parameter

--parameter=value

-p

-p value

4

FFmpeg est utilisé partout

- VLC
- mpv
- Audacity
- Handbrake
- Shutter Encoder
- AEO-Light
- vrecord
- QCTools
- Google Chrome
- YouTube
- Adobe Premiere
- DaVinci Resolve
- ...

5

La famille FFmpeg

programmes

- ffmpeg
- ffprobe
- ffplay

bibliothèques

- libavformat
- libavcodec
- libavfilter
- libavutil
- libavdevice
- libswscale
- libswresample

6

Conversions de fichiers

ffmpeg (CLI)

→ ffmpeg.org

FFmpeg Cookbook for Archivists

→ avpres.net/FFmpeg/

ffmprovisr

→ amiaopensource.github.io/ffmprovisr/

7

Extraction des métadonnées

MediaInfo (GUI, CLI)

→ mediaarea.net/MediaInfo

ffprobe (CLI)

→ ffmpeg.org

8

Lecteurs multimédia

VLC (GUI)

→ www.videolan.org/vlc/

mpv (CLI/GUI)

→ mpv.io

ffplay (CLI)

→ ffmpeg.org

9

Interfaces graphiques

- ...
- Audacity
- Handbrake
- Shutter Encoder

10

Structure de commandes FFmpeg

\$0 **\$1** **\${n}**
command argument_1 ... argument_n

Syntaxe des arguments:

-parameter
-parameter value
-p
-p value

11

Syntaxe de FFmpeg

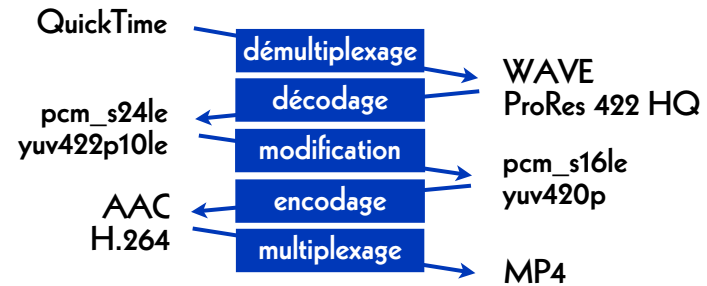
ffmpeg
[global_options]
[input_options_n] -i input_file_n
[output_options_n] output_file_n

ffprobe *[input_options] input_file*

ffplay *[input_options] input_file*

12

Exemple: image et son



13

Bibliothèques utilisées

démultiplexage: libavformat
décodage: libavcodec
modification: libavfilter
encodage: libavcodec
multiplexage: libavformat

14

Initiation

15

Définir le dossier de travail

```
# Linux  
# Mac  
# Windows WSL ou Terminal  
# Windows localement  
cd Desktop  
  
# Windows avec OneDrive Cloud  
cd OneDrive
```

16

Générer un fichier vidéo

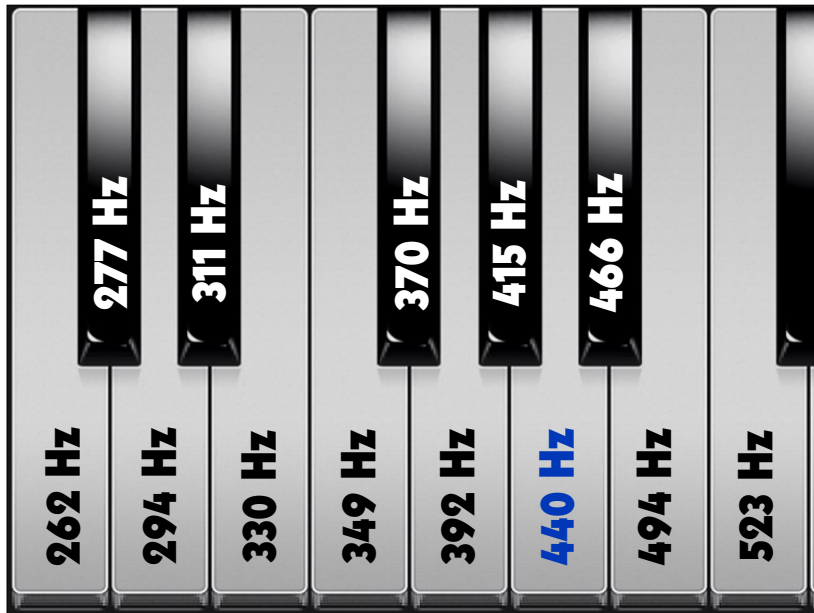
```
ffmpeg  
-f lavfi -i mandelbrot  
-t 10  
-c:v rawvideo  
-pix_fmt uyvy422  
mandelbrot.avi
```

17

Lire le fichier vidéo

```
ffplay  
mandelbrot.avi
```

18



19

Générer un fichier audio

```
ffmpeg  
-f lavfi -i "sine=frequency=440"  
-t 10  
-c:a pcm_s16le  
-ar 48k  
-ac 2  
la.wav
```

Exercice: utiliser d'autres valeurs que 440, comme par exemple 262, 330, 392 ou 523

20

Lire le fichier audio

```
ffplay  
la.wav
```

21

Assembler les fichiers

```
ffmpeg  
-i mandelbrot.avi  
-i la.wav  
-c:v copy  
-c:a copy  
mandela.avi
```

22

Lire le fichier

```
ffplay  
mandela.avi
```

23

Extraire les métadonnées

```
ffprobe  
mandela.avi
```

24

Conteneur

```
ffprobe  
-show_format  
mandela.avi
```

25

Codec

```
ffprobe  
-show_streams  
mandela.avi
```

26

Conteneur et codec

```
ffprobe  
-show_format  
-show_streams  
mandela.avi
```

27

Formater les métadonnées

```
ffprobe  
-show_format  
-show_streams  
-print_format json  
mandela.avi
```

Exercice: utiliser flat et xml au lieu de json

28

Enregistrer les métadonnées

```
ffprobe  
-show_format  
-show_streams  
-print_format json  
mandela.avi  
> mandela.txt
```

29

Extraire les métadonnées

```
mediainfo  
mandela.avi
```

30

Chercher de l'aide

```
ffmpeg -h  
ffmpeg -codecs  
ffmpeg -decoders  
ffmpeg -h decoder=aac  
ffmpeg -encoders  
ffmpeg -h encoder=libx264  
ffmpeg -filters  
ffmpeg -h filter=scale  
ffmpeg -pix_fmts  
ffmpeg -sample_fmts
```

31

Conversionns

32

Changer le conteneur

```
ffmpeg  
-i mandelbrot.avi  
-c copy  
mandelbrot.mov
```

33

Somme de contrôle (1)

```
ffmpeg  
-i mandelbrot.avi  
-f framemd5  
mandelbrot.avi_framemd5.txt
```

34

Somme de contrôle (2)

```
ffmpeg  
-i mandelbrot.mov  
-f framemd5  
mandelbrot.mov_framemd5.txt
```

35

Comparer les sommes de contrôle

```
# Linux/Mac/Windows WSL ou Terminal  
diff -s  
mandelbrot.avi_framemd5.txt  
mandelbrot.mov_framemd5.txt  
  
# Windows  
fc  
mandelbrot.avi_framemd5.txt  
mandelbrot.mov_framemd5.txt
```

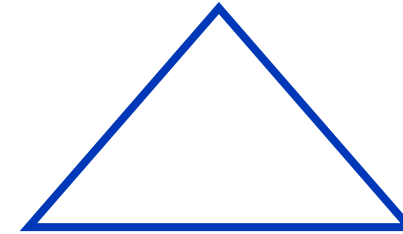
36

Lecture d'images individuelles

```
ffplay  
-loop 0  
DUFAY_TIFF/Dufay_#06d.tif
```

37

qualité de l'image



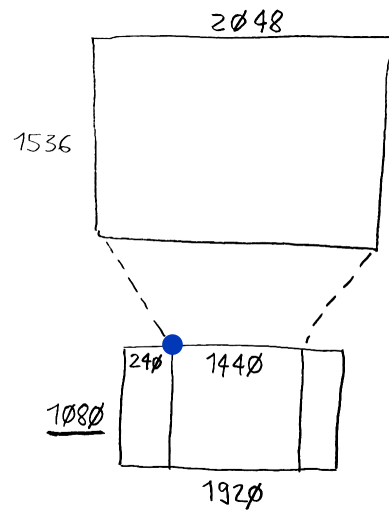
temps d'encodage taille du fichier

38

2K
4:3



HD
16:9

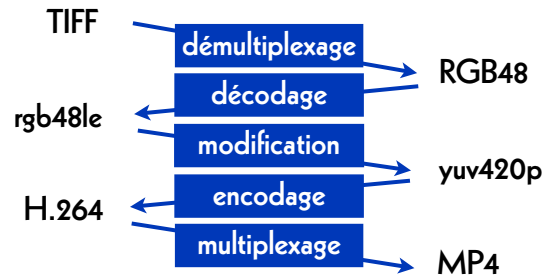


39

H.264

40

De TIFF à H.264



41

Archive → Accès

```
ffmpeg
-f image2 -framerate 24
-i DUFAY_TIFF/Dufay_%06d.tif
-filter:v
  "scale=1440:1080:flags=lanczos,
  pad=1920:1080:240:0"
-c:v libx264 -preset veryslow -crf 21
-pix_fmt yuv420p
-movflags +faststart+write_colr
Dufay_21_H264.mp4
```

42

Contrôle de qualité

43

Contrôle de qualité

- vérifier les métadonnées techniques
- analyser les signaux
- visualiser et/ou écouter le fichier

- écran divisé entre deux fichiers
- différentiel entre deux fichiers

44

Outils pour le contrôle de qualité

conteneur et codec

- MediaInfo, ffprobe, MediaConch
- hexdump, fq

contenu de l'image et du son

- QCTools, qcli, SignalServer
- VLC, mpv, ffplay, GridPlayer

45

Écran divisé

ffmpeg

```
-i Dufay_21_H264.mp4  
-i Dufay_35_H264.mp4  
-filter_complex  
    "[0] crop=iw/2:ih:0:0 [left];  
    [1] crop=iw/2:ih:iw/2:0 [right];  
    [left][right] hstack"  
Dufay_split.mp4
```

46

Tester un filtre

ffplay

-vf "negate"

Dufay_21_H264.mp4

47

Différentiel

ffmpeg

```
-i Dufay_21_H264.mp4  
-i Dufay_35_H264.mp4  
-filter_complex  
    "[1] format=yuva444p,  
    lut=c3=128,  
    negate [1_with_alpha];  
    [0][1_with_alpha] overlay"  
Dufay_delta.mp4
```

48

Générer un fichier audio

```
ffmpeg  
-f lavfi -i "anoisesrc=color=brown"  
-filter:a "tremolo=f=0.1:d=0.9"  
-c:a pcm_s24le  
-ar 96k  
-ac 2  
-t 60  
seashore_good.wav
```

49

Lire le fichier audio

```
ffplay  
seashore_good.wav
```

50

Endommager le fichier audio

```
ffmpeg  
-i seashore_good.wav  
-c copy  
-bsf:a "noise=amount=-1"  
seashore_bad.wav
```

51

Lire le fichier audio endommagé

```
ffplay  
seashore_bad.wav
```

52

Visualiser le volume (fichier intact)

```
ffplay
-f lavfi
  "amovie=seashore_good.wav,
    asplit [a][out1];
[a] showvolume=c=VOLUME:
  w=1000:h=100:
  ds=lin [out0]"
```

53

Visualiser le volume (fichier endommagé)

```
ffplay
-f lavfi
  "amovie=seashore_bad.wav,
    asplit [a][out1];
[a] showvolume=c=VOLUME:
  w=1000:h=100:
  ds=lin [out0]"
```

54

Visualiser l'onde sonore (fichier intact)

```
ffplay
-f lavfi
  "amovie=seashore_good.wav,
    asplit [a][out1];
[a] showwaves=mode=cline [out0]"
```

Exercice: utiliser point ou p2p au lieu de cline

55

Visualiser l'onde sonore (fichier endommagé)

```
ffplay
-f lavfi
  "amovie=seashore_bad.wav,
    asplit [a][out1];
[a] showwaves=mode=cline [out0]"
```

Exercice: utiliser point ou p2p au lieu de cline

56

Visualiser le spectrogramme (fichier intact)

```
ffplay  
-f lavfi  
  "amovie=seashore_good.wav,  
    asplit [a][out1];  
[a] showspectrum=mode=separate:  
  color=intensity:  
  slide=1:  
  scale=cbrt [out0]"
```

57

Visualiser le spectrogramme (fichier endommagé)

```
ffplay  
-f lavfi  
  "amovie=seashore_bad.wav,  
    asplit [a][out1];  
[a] showspectrum=mode=separate:  
  color=intensity:  
  slide=1:  
  scale=cbrt [out0]"
```

58

Personnaliser

59

Vérifier les polices installées

```
# Linux/Mac/Windows WSL ou Terminal  
ls /Library/Fonts
```

```
# Ubuntu  
fc-list
```

```
# Windows  
dir \Windows\Fonts
```

60

Insérer un filigrane

```
ffmpeg
-i Dufay_21_H264.mp4
-filter:v
  "drawtext=text='watermark':
    fontfile='/Library/Fonts/Arial.ttf':
    fontsize=35:
    fontcolor=white:
    alpha=0.25:
    x=(w-text_w)/2:y=(h-text_h)/2"
with_watermark.mp4
```

61

Insérer un timecode

```
ffmpeg
-i Dufay_21_H264.mp4
-filter:v
  "drawtext=timecode='01\:00\:30\:00':
    rate=25:
    fontfile='/Library/Fonts/Arial.ttf':
    fontsize=35:
    fontcolor=white:
    x=(w-text_w)/2:y=h/1.2"
with_timecode.mp4
```

62

Insérer un logo

```
ffmpeg
-i Dufay_21_H264.mp4
-i logo.png
-filter_complex
  "overlay=10:main_h-overlay_h-10"
with_logo.mp4
```

63

AV Preservation by
reto.ch

Sandrainstrasse 3/7
3007 Bern
Switzerland

reto.ch
info@reto.ch



64